From Sentence to Concept

Anne Preller

▶ **To cite this version:**

Anne Preller. From Sentence to Concept. E. Grefenstette, C. Heunen, and M. Sadrzadeh. Categorical Information Flow in Physics and Linguistics, Oxford University Press, pp.247–271, 2012. lirmm-00635866v1

HAL Id: lirmm-00635866

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00635866v1

Submitted on 26 Oct 2011 (v1), last revised 22 Sep 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Sentence to Concept

Predicate Logic and Quantum Logic in Compact Closed Categories

Anne Preller *

Abstract

The compositional functional logical models of natural language are recast as compact closed categories. Composition is based on the geometrical representation of information flow characteristic for these categories. The functional logical interpretation of (strings of) words is carried over to projectors in a finite tensor product of 2-dimensional spaces such that the truth of a sentence is equivalent to the truth of the corresponding projector.

Examples include sentences with compound noun phrases involving quantifiers, adjectives and negation.

Keywords: Compact closed categories, quantum logic, concept spaces, two-sorted first order logic, compositional semantics, pregroup grammars, proof graphs, compound noun-phrases

1 Introduction

The present work attempts to relate two semantic representations of natural language, the functional logical models and the distributional vector models. The former deals with individuals and their properties, the latter with concepts and how they can be approximated.

Montague semantics and similar functional logical models for natural language are extensional and compositional. Meaningful expressions designate individuals, sets of individuals, functions from and to (sets of) individuals, truth-value functions and so on. The meaning of a grammatical string of words is computed from the meanings of the constituents using functional application or composition. This semantics requires prior grammatical analysis where every word contributes to the meaning, including 'noise' like negation, determiners, quantifiers, relative pronouns , etc.

The semantic vector models are based on the principle that the content of a word is measured in relation to the content of other words. They handle probabilistic estimations of concepts. Words, with the exclusion of 'noise', are represented by vectors in a finite dimensional space over the field of real numbers. Frequency counts of co-occurrences with other words determine the coordinates

*LIRMM, 161 rue Ada, 34194 Montpellier, France. E-mail: preller@lirmm.fr

of a word. Semantic vector models excel in detecting similarity of words. They confound opposites.

Compositionality of vector semantics remains an open question and is subject of intensive research.

One approach to compositionality is quantum logic on the lattice of projectors of Hilbert spaces, see [Rijsbergen, 2004] for an overview oriented towards information retrieval or [Widdows, 2004] for a discussion of geometric properties of meaning. There is, however, no general algorithm that transforms a string of word into a vector respecting the logic. Another approach is composition of vectors by the tensor product, [Smolensky, 1988] invoking computational principles of cognitive science, or [Clark and Pulman, 2007] and [Clark et al., 2008] using syntactical analysis. Again, 'noise' is not included in composition.

The present work outlines a method that takes into account the logical content of 'noise' and transforms the compositional extensional representation into a conceptual representation. Both representations are based on biproduct dagger compact closed categories.

On one hand, a *concept space*, that is to say a tensor product of two-dimensional spaces, hosts both the words (concepts) and their probabilistic approximations. Concept spaces are the linguistic pendent to compound systems in quantum mechanics.

On the other hand, the logical functional representations of (strings of) words are also recast as vectors. These vectors are, roughly speaking, the names of the functions representing the words. Their construction involves syntactical analysis by a pregroup grammar, [Lambek, 1999].

Pregroup calculus, also known as compact bilinear logic, [Lambek, 1993], is a simplification of the syntactic calculus by the same author, [Lambek, 1958]. Compact bilinear logic 'compacts' the higher order of categorial grammars into second order logic with general models, or equivalently, into two-sorted first order logic, [Benthem and Doets, 1983]. Moreover, the category of types and proofs of compact bilinear logic is the free compact *2*-category , [Preller and Lambek, 2007].

Categorical semantics in compact 2-categories for pregroup grammars was first proposed in [Preller, 2005], reformulated in [Preller, 2007] in terms of functions in two-sorted first order logic. This reformulation rests on the fact that sets and two-sorted functions form a compact closed category. The embedding of the category of two-sorted functions in the category of semi-modules over a real interval, [Preller and Sadrzadeh, 2011], establishes the connection to semantic vector models.

The formulation of functional logic in a biproduct dagger compact closed category has been chosen to facilitate a comparison with quantum logic. It is based on [Abramsky and Coecke, 2004], casting quantum mechanics in the abstract setting of a biproduct dagger compact closed category. The result is an embedding of functional two-sorted first order logic into the lattice of projectors of concept spaces.

Section (2) introduces the semantical and syntactical categories. The category of two-sorted functions follows in Section 3 with its two-sorted first order

logic. An embedding transfers them to an arbitrary bicategory dagger compact closed category.

The algorithm in Section 4 composing meanings of strings from meanings of words is based on syntactical analysis. Examples from natural language provide the graphs depicting the computation of the meaning by 'information flow'.

Concept spaces and the logical properties of their intrinsic projectors are investigated in Section 5. Subsections 5.1 and 5.2 deal with propositional logic and predicate logic. The truth preserving one-one correspondence between predicates on and intrinsic projectors of concept spaces is the subject of 5.3. This correspondence is used in Section 5.4 to compute the meaning of strings directly in concept spaces and to view arbitrary word vectors as a probabilistic approximation of concepts.

2 Notations, basic properties

Natural language processing involves both syntactical analysis and logical representation. Both can be formulated in the language of compact bicategories, also known as non-symmetric star autonomous categories.

Throughout this paper, the *syntactical category* is a freely generated compact bicategory. It is not symmetric.

The *semantic category* $\mathcal{C}$ is any biproduct dagger compact closed category in which all objects have a chosen finite basis, for example the category $\mathcal{IR}$ of free semi-modules generated by finite sets over the lattice of the real interval $[0,1]$.

2.1 The syntactical category

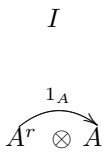
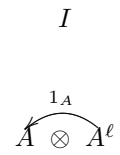
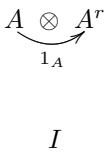
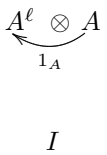
The syntactical category $\mathcal{C}2(\mathcal{B})$ is the free compact bicategory generated by a category $\mathcal{B}$. It is notationally convenient to replace the canonical associativity and unit isomorphisms by identities, for example $A \otimes (B \otimes C) = (A \otimes B) \otimes C$, $A \otimes I = A = I \otimes A$. Strictly speaking, the bicategory is treated like a 2-category.

Saying that aa bicategory is compact means that every 1-cell A has a left adjoint A^ℓ and a right adjoint A^r. Let $\eta_A : I \to A^r \otimes A$ be the unit and $\epsilon_A : A \otimes A^r \to I$ the counit for the right adjoint. Then $A \simeq A^{\ell r}$ is a right adjoint to A^ℓ so that $\eta_{A^\ell} : I \to A \otimes A^\ell$ and $\epsilon_{A^\ell} : A^\ell \otimes A \to I$ act as unit and counit for the right adjunction of A to A^ℓ and to the left adjunction of A_ℓ to A.

Starting with any 1-cell A that is an object of $\mathcal{B}$, one obtains the *iterated right* adjoints $A^\ell, A^{\ell\ell}, A^{\ell\ell\ell}, \ldots$ and the *iterated left* adjoints $A^r, A^{rr}, A^{rrr}, \ldots$ of A, but no mixed adjoints, because $A^{\ell r}$ and $A^{r\ell}$ are both isomorphic to A.

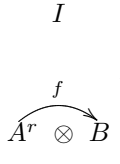
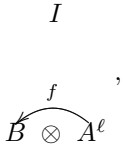
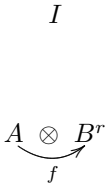
The *morphisms*, i.e. the 2-cells of $\mathcal{C}2(\mathcal{B})$, are represented by graphs where the vertices are objects of $\mathcal{B}$ and the oriented links are labelled by morphisms

of $\mathcal{B}$. Examples are

$\eta_A =$  , $\eta_{A^\ell} =$  , $\epsilon_A =$  , $\epsilon_{A^\ell} =$ 

NOTE: *graphs display the domain of the morphism above, the codomain below.*

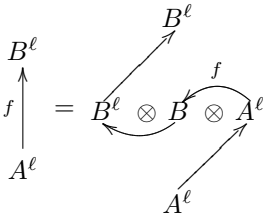
In the case where the label is an identity, it is in general omitted. An arbitrary $f : A \to B \in \mathcal{B}$ also creates labels for links, for example

$\ulcorner f \urcorner =$  , $(f \otimes 1_{A^\ell}) \circ \eta_{A^\ell} =$  , $\llcorner f \lrcorner =$  , etc.

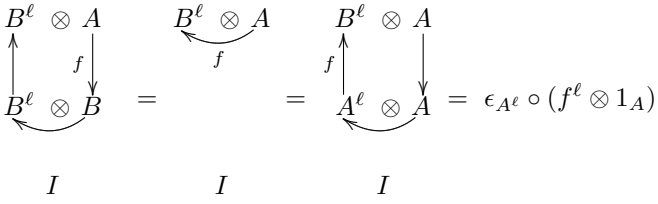
NOTE: *The labels of the links are morphisms of $\mathcal{B}$. Stripping the tail of the link of its adjoints, we obtain the domain of the label in $\mathcal{B}$. Similarly, the head without the adjoints is the codomain of the label.*

Composition of morphism is computed by connecting the graphs at the joint interface and walking paths, picking up and composing the labels in the order in which they appear.

Here are few examples where $f : A \to B$, $g : B \to C$

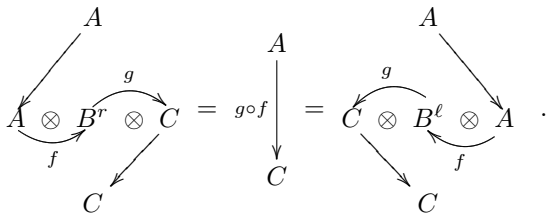
$f^\ell =$  $= (\epsilon_{B^\ell} \otimes 1_{A^\ell}) \circ (1_{B^\ell} \otimes ((f \otimes 1_{A^\ell}) \circ \eta_{A^\ell}))$

Recall: the domain of the morphism f^ℓ represented by the graph is the top line, the codomain is the bottom line, i.e. $f^\ell : B^\ell \to A^\ell$.

$\epsilon_{B^\ell} \circ (1_{B^\ell} \otimes f) =$  $= \epsilon_{A^\ell} \circ (f^\ell \otimes 1_A)$

An equality of graphs is far easier to compute than the equality of the corresponding algebraic expressions. For example, the equality $(\llcorner f \lrcorner \otimes 1_C) \circ (1_A \otimes$

$\ulcorner g \urcorner) = g \circ f = (1_C \otimes (\epsilon_{B^\ell} \circ (1_{B^\ell} \otimes f))) \circ (((g \otimes 1_{B^\ell}) \circ \eta_{B^\ell}) \otimes 1_A) : A \to C$ is proved thus



NOTE: *Links do not cross in the graphs representing morphisms of the syntactical category.*

The benefit of orienting and labelling links will become evident through the examples of natural language processing in Section 4.1.

2.2 The semantic category

The general definitions and properties of biproduct dagger compact closed categories can be found in [Selinger, 2007]or [Abramsky and Coecke, 2004]. Here they are used in particular in $\mathcal{IR}$, which is tailored to natural language semantics.

Denote $\mathcal{IR}$ the category of free semi-modules over the lattice of the real interval $[0, 1]$ generated by finite sets. Its importance to natural language processing resides in the fact that the coordinates of word vectors are obtained by frequency counts of co-occurrences of words in text-windows.

Recall that the linear order on the real numbers in $[0, 1]$ induces a distributive and implication-complemented lattice structure on $[0, 1]$, namely

$$\alpha \vee \beta = \max\{\alpha, \beta\} \text{ and } \alpha \wedge \beta = \min\{\alpha, \beta\}$$
$$\alpha \to \beta = \max\{\gamma \in I : \ \alpha \wedge \gamma \leq \beta\}$$
$$\neg\alpha = \alpha \to 0\,.$$

This lattice is not Boolean, because $\neg\neg\alpha = 1 \neq \alpha$ for $0 < \alpha < 1$. The set $\{0, 1\}$, however, forms a Boolean algebra.

The lattice operations define a semiring structure on $[0, 1]$ with neutral element 0 and unit 1 by

$$\alpha + \beta = \alpha \vee \beta \qquad \alpha \cdot \beta = \alpha \wedge \beta\,.$$

Basic properties

Objects of an arbitrary biproduct dagger compact closed category are called *spaces*, morphisms *linear maps*, and linear maps $v : I \to V$ *vectors of* V. Write $v \in V$ for vectors $v : I \to V$ and $f(v) \in W$ for $f \circ v : I \to W$, where $f : V \to W$.

Vectors $b_1, \ldots, b_n$ of V form a *basis* of V if every vector of V can be written in a unique way as a linear combination of the vectors $b_1, \ldots, b_n$. A space is

n-dimensional if it has a basis of cardinality n. The dimension is unique. A space with chosen basis $B = b_1, \ldots, b_n$ is denoted V_B.

All spaces are assumed to be finite dimensional from now on.

Linear maps identify with matrices such that multiplication of matrices corresponds to composition of maps. Indeed, let $A = a_1, \ldots, a_m$ and $B = b_1, \ldots, b_n$. A linear map $f : V_A \to V_B$ is determined by its values on the basis vectors $a_1, \ldots, a_m \in A$ and is characterized by the matrix (ϕ_{ij}) where $\phi_{ij} \in I$ is the i-th coordinate of $f(a_j) = \sum_{i=1}^{n} \phi_{ij} b_i$. Such matrices can be identified with vectors in $V_A \otimes W_B$, because the tensor product $V_A \otimes W_B$ has basis vectors $a_i \otimes b_j$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

The *inner product* $\langle v|w\rangle$ of vectors $v = \sum_{j=1}^{m} \alpha_j a_j$ and $u = \sum_{j=1}^{m} \beta_j a_j$ in $\mathcal{RI}$ is given by

$$\langle v|u\rangle = \langle \sum_{j=1}^{m} \alpha_j a_j | \sum_{j=1}^{m} \beta_j a_j \rangle = \sum_{j=1}^{m} \alpha_j^\dagger \beta_j \,.$$

Vectors are *orthogonal* if $\langle v|u\rangle = 0$. In the case of $\mathcal{RI}$, we have $\alpha = \alpha^\dagger$ for all scalars $\alpha \in [0, 1]$. Hence vectors with coordinates in $[0, 1]$ are orthogonal in $\mathcal{RI}$ exactly when they are orthogonal in the category of Hilbert spaces.

The category of semimodules $\mathcal{IR}$ is a biproduct dagger compact closed category with monoidal unit $I = [0, 1]$. Every object V of $\mathcal{RI}$ has a unique finite basis A, which we express by $V = V_A$. It is its own adjoint, $V_A = V_A^*$. The unit $\eta_{V_A} : I \to V_A \otimes V_A$ and counit $\varepsilon_{V_A} : V_A \otimes V_A \to I$ of the adjunction are given by

$$\eta_{V_A}(1) = \sum_{a \in A} a \otimes a \qquad \varepsilon_{V_A}(a \otimes b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{else} \end{cases} = \langle a|b\rangle \,.$$

The name and coname of $f : V_A \to V_B$ are defined by

$$\begin{array}{l} \ulcorner f \urcorner(1) = \sum_{a \in A} a \otimes f(a) \\ \llcorner f \lrcorner(a \otimes b) = \langle f(a)|b\rangle \text{ for } a \in A,\, b \in B \,. \end{array}$$

By definition, $V_A = V_A^\dagger$. The adjoint of $f : V_A \to V_B$ is the morphism $f^* = f^\dagger$ induced by the transpose of the matrix of f.

The logic of vectors

Vectors of any space V_B in $\mathcal{RI}$ are partially ordered by the product order of [0,1]. The *null vector* $\overrightarrow{0}$ (with coordinates all equal to 0) is the smallest and the *full vector* $\overrightarrow{1}$ (with coordinates all equal to 1) the largest vector. Accordingly, the logical connectives defined on [0,1] introduce logical connectives on V_B, defined coordinate by coordinate

$$\neg \sum_{i=1}^{n} \alpha_i b_i = \sum_{i=1}^{n} (\neg \alpha_i) b_i, \;\; (\sum_i \alpha_i b_i) \wedge (\sum_j \beta_j b_j) = \sum_i (\alpha_i \wedge \beta_i) b_i \text{ etc.} \qquad (1)$$

Lemma 1. *The vector connectives define a distributive, implication complemented lattice structure on V_B such that the following equivalences hold*

$$\neg\neg v = v \iff v \vee \neg v = \overrightarrow{1} \iff \textit{the coordinates of } v \textit{ are } 0 \textit{ or } 1\,. \qquad (2)$$

Definition 1 (Boolean vector). *A vector $v = \sum_{i=1}^{n} \alpha_i b_i$ is* Boolean *if $\alpha_i \in \{0,1\}$, for $i = 1, \ldots, n$.*

Vector connectives can be defined on Boolean vectors in any semantic category. The scalars $0, 1$ identify with $0_{0I} : 0 \to I$ and $1_I : I \to I$. The connectives $\neg$, $\wedge$ etc. are explicitly defined operators on the set $\{0,1\}$ by

$$\neg 0 = 1, \neg 1 = 0 \text{ and } 1 \wedge 1 = 1, 0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0, \text{etc.}$$

They lift to the Boolean vectors as indicated in (1). The following properties hold in all semantic categories

1. the Boolean vectors together with the logical vector connectives form a Boolean algebra
2. every Boolean v defines a unique subset $K \subseteq B$ such that $v = \sum_{b \in K} b$ and vice versa
3. the linear map $\wedge : V_B \otimes V_B \to V_B$ satisfying

$$\wedge(b \otimes b) = b, \quad \wedge(b \otimes b') = \overrightarrow{0}\,, \text{ for } b \neq b' \in B$$

coincides with vector conjunction on all Boolean vectors.

In $\mathcal{RI}$, vector conjunction coincides everywhere with the linear map defined in Item 3.

The logic of projectors

Let $\mathcal{C}$ be any semantic category and E an n-dimensional space with chosen basis $B = b_1, \ldots, b_n$.

Recall that a morphism $p : E \to E$ is a *projector* if it is idempotent and self-adjoint

$$p \circ p = p, \quad p^\dagger = p\,.$$

In $\mathcal{RI}$, the latter equality means that the matrix of p is symmetric.

Every projector p determines a subspace, namely the set of vectors invariant under p

$$E_p = \{w \,:\, w = p(w)\} = p(E)\,.$$

The subspaces of Hilbert spaces are in one-one correspondence with projectors. Hence the quantum connectives are defined on the set of projectors/subspaces thus

$$\neg E_p = E_p^\perp,\ E_p \vee E_q = E_p + E_q,\ E_p \wedge E_q = E_{p \circ q},$$
$$E_p \to E_q = \{u \,:\, q(p(u)) = p(u)\}\,.$$

They induce a lattice structure on the set of projectors such that $p \leq q$ is equivalent to $p \wedge q = q$, see [Rijsbergen, 2004]. Thinking of projectors as propositions

and of 1_E as the true proposition, the equality $p \to q = 1_E$ is read as 'p implies q'. This lattice is not distributive in general.

This approach is not possible in the general case. A two-dimensional space of $V_{\{a_1,a_2\}}$ in $\mathcal{IR}$ has subspaces that are not image of any projector. An example is the subspace generated by the vectors $u = \alpha a_1$, $v = \beta a_2$, $w = \gamma a_1 + \beta a_2$, where $0 < \beta < \alpha < \gamma \leq 1$. We need a property that connects subspaces and projectors in an arbitrary semantic category.

Definition 2 (**Intrinsic morphism**). *A linear map of $\mathcal{C}$ is* intrinsic *if it sends every basis vector to a basis vector or to the null vector.*

Intrinsic linear maps are closed under composition.

Lemma 2. *A projector $p : E \to E$ is intrinsic if and only if*

$$p(b_i) = b_i \text{ or } p(b_i) = \overrightarrow{0}, \text{ for } i = 1, \ldots, n\,. \tag{3}$$

Proof. Let p be an intrinsic projector and (π_{ij}) its matrix. This matrix is symmetric, because p is self-adjoint and the entries π_{ij} are 0 or 1.

We must show that $p(b_k) = b_l$ implies $k = l$. From $p(b_k) = b_l$ follows $p(b_l) = b_l$, because p is idempotent. The latter equality implies $\pi_{ll} = 1$ and $\pi_{il} = 0$ for $i \neq l$. Moreover, $p(b_k) = b_l$ implies $\pi_{lk} = 1$ and $\pi_{ik} = 0$ for $i \neq l$. By symmetry, $\pi_{kl} = 1$. Hence $k = l$. □

One consequence is that every intrinsic projector has the form $\sum_{k\in K} |b_k\rangle\langle b_k|$ where $K \subseteq \{1, \ldots, n\}$. Recall that $|b_i\rangle\langle b_i|$ maps b_i to itself and every other basis vector to the null vector, for $i = 1, \ldots, n$.

Intrinsic projectors map Boolean vectors to Boolean vectors. The composition $q \circ p$ of intrinsic projectors $p : E \to E$ and $q : E \to E$ is again an intrinsic projector satisfying

$$(q \circ p)(x) = q(x) \wedge p(x), \text{ for all } x \in B\,.$$

Intrinsic projectors behave well: there is a one-one correspondence between intrinsic projectors, subspaces generated by a subset of basis vectors and Boolean vectors.

Let $v = \alpha_1 b_1 + \cdots + \alpha_n b_n$ be a Boolean vector of E. Define the linear map $p_v : E \to E$ by its values on the basis vectors

$$p_v(b_i) = \alpha_i b_i, \text{ for } i = 1, \ldots, n\,. \tag{4}$$

Remark 1.

1. p_v is an intrinsic projector
2. $p_v(w) = v \wedge w$ for every Boolean vector w; in particular $p_v(\overrightarrow{1}) = v$
3. the map $v \mapsto p_v$ is one-one
4. $p_{\overrightarrow{1}} = 1_E$

Lemma 3. *For every intrinsic projector p there is a Boolean vector v such that $p = p_v$. For every Boolean vector v, the subspace E_{p_v} of vectors invariant under p_v coincides with the subspace generated by the basis vectors b_i satisfying $b_i \leq v$.*

Proof. Let p be an intrinsic projector. Define

$$K = \{k \,:\, p(b_k) = b_k \,\&\, 1 \leq k \leq n\} \text{ and } v = \sum_{k \in K} b_k \,.$$

Then $p(b_i) = p_v(b_i)$, for $i = 1, \ldots, n$. Hence the map $v \mapsto p_v$ is onto the intrinsic projectors of E. Moreover, E_p is generated by the set of basis vectors $\{b_k : k \in K\}$ and $b_i \leq v$ if and only if $i \in K$. □

Theorem 1. *The map $v \mapsto p_v$ is a negation preserving lattice isomorphism $\mathcal{H}$ from the Boolean vectors onto the intrinsic projectors of E such that $\mathcal{H}(\overrightarrow{1}) = 1_E$.*

Proof. Writing $E_v = E_{p_v}$, we must prove that

$$E_v{}^{\perp} = E_{\neg v},\ E_v \vee E_w = E_{v \vee w},\ p_v \circ p_w = p_{v \wedge w},\ E_v \to E_w = E_{v \to w} \,.$$

The first two equalities are straight forward. The third equality $p_v \circ p_w = p_{v \wedge w}$ is equivalent to $p_v \circ p_w(\overrightarrow{1}) = p_{v \wedge w}(\overrightarrow{1})$, which follows from Remark 1. To prove the last equality, let $v = \sum_{i=1}^n \alpha_i b_i$, $w = \sum_{i=1}^n \beta_i b_i$ be Boolean vectors. Then $v \to w = \sum_{i=1}^n (\neg\alpha_i \vee \beta_i) b_i$. Let $u = \sum_{i=1}^n \gamma_i b_i \in E_{v \to w}$ be arbitrary. Then $u = p_{v \to w}(u)$. The latter equality is equivalent to the condition 'if $\alpha_i = 1$ and $\beta_i = 0$ then $\gamma_i = 0$, for $i = 1, \ldots, n$'. On the other hand, $u \in E_v \to E_w = \{u : p_w(p_v(u)) = p_v(u)\}$ if and only if $\alpha_i\gamma_i = \alpha_i\beta_i\gamma_i$ for $i = 1, \ldots, n$. □

It follows that the sublattice of intrinsic projectors is Boolean. Moreover, intrinsic projectors are monotone increasing on Boolean vectors.

If E is a space of $\mathcal{IR}$ the equalities (4) define a projector for every vector v of E. The intrinsic projectors are exactly those defined by Boolean vectors.

3 Two-sorted first order logic in compact closed categories

The relevance of two-sorted first order logic for natural language is due to the fact that it is equivalent to second order logic with general models, see [Benthem and Doets, 1983], and a wide spread belief that second order logic suffices for natural language semantics.

3.1 The category of two-sorted functions

Two-sorted functions are tailored to natural language in the sense that they accept both elements (sort 1) and sets (sort 2) as arguments. In a similar way, verbs accept both singulars and plurals. Functions in two-sorted first order logic were introduced in [Preller, 2007]. The presentation below follows [Preller and Sadrzadeh, 2011].

Definition 3 (**Two-sorted function**). *A function $f : A \to B$ is* two-sorted *if it maps elements and subsets of A to elements or subsets of B and satisfies*

$$\begin{array}{rcl} f(\{a\}) & = & f(a) \text{ for } a \in A \\ f(\emptyset) & = & \emptyset \\ f(X \cup Y) & = & f(X) \cup f(Y) \text{ for } X, Y \subseteq A\,. \end{array} \tag{5}$$

Obviously, a two-sorted function defined on a finite set is determined by its values on the elements. An example is the *two-sorted identity*

$$\begin{array}{l} 1_A(a) \;= a, \text{ for } a \in A \\ 1_A(X) = X, \text{ for } X \subseteq A\,. \end{array}$$

Lemma 4. *The category $2\mathcal{SF}$ of finite sets and two-sorted functions is a dagger biproduct compact closed category.*

Proof. The biproduct is the disjoint union of sets. Hence $V_A = A$ for every finite set A.

The monoidal structure is given by the cartesian product of sets. A two-sorted notation for the Cartesian product brings the same notational advantages as the tensor product

$$\begin{array}{ll} a \times_2 b & = \langle a, b \rangle \\ a \times_2 B & = \{a\} \times B \\ A \times_2 b & = A \times \{b\} \\ A \times_2 B & = A \times B\,. \end{array}$$

The two-sorted product $f \times_2 g : A \times_2 C \to B \times_2 D$ for $f : A \to B$ and $g : C \to D$ is determined by it values on the elements of $A \times_2 C$, namely

$$(f \times_2 g)(a \times_2 b) = f(a) \times_2 g(b), \text{ for } a \in A,\, b \in C\,.$$

The monoidal unit is the singleton set $I = \{a_0\}$.

Every object is its own adjoint, $A = A^* = A^\dagger$, the unit $\eta_A : I \to A \times A$ and counit $\varepsilon_A : A \times A \to I$ of the adjunction are given by

$$\eta_A(a_0) = \{a \times_2 a \,:\, a \in A\} \qquad \varepsilon_A(a \times_2 b) = \begin{cases} a_0 & \text{if } a = b \\ \emptyset & \text{else} \end{cases}$$

The name, coname, dual and dagger of $f : A \to B$ are given by

$$\begin{array}{l} \ulcorner f \urcorner(a_0) = \{a \times_2 b \,:\, f(a) = b \text{ or } b \in f(a),\, a \in A,\, b \in B\} \\ \llcorner f \lrcorner(a \times_2 b) = \begin{cases} a_0 & \text{if } f(a) = b \text{ or } b \in f(a) \\ \emptyset & \text{else} \end{cases} \\ f^*(b) = \{a \in A \,:\, f(a) = b \text{ or } b \in f(a)\} = f^\dagger(b)\,. \end{array}$$

□

The category $2\mathcal{SF}$ has an abundance of projectors. Here is an example.

Example 1. *A two-sorted projector that is not intrinsic.*

The two-sorted function $p : \{a, b\} \to \{a, b\}$ defined by

$$p(a) = \{a, b\} \text{ and } p(b) = \{a, b\}$$

is a projector in $2\mathcal{SF}$. The corresponding matrix is

$$(\pi_{ij}) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} .$$

The same matrix induces an endomorphism in any two-dimensional space. It is again a projector in $\mathcal{RI}$, but not in a Hilbert space, because it is not idempotent when viewed as a linear map of Hilbert spaces.

Luckily, the semantics of natural language only involves intrinsic projectors which live in every semantic category. The characterization of intrinsic projectors (3) is equivalent in $2\mathcal{SF}$ to

$$p(Y) = \{x \in B \,:\, p(x) = x\} \cap Y, \text{ for every } Y \subseteq B\,. \tag{6}$$

3.2 The embedding

Given a set $A = \{a_1, \ldots, a_m\}$, define a map $\mathcal{J}_A$ defined for elements and subsets of A with values in the space V_A by declaring for $X \subseteq A$ and $a \in A$

$$\mathcal{J}_A(X) = \sum_{a \in X} a, \quad \mathcal{J}_A(a) = a\,. \tag{7}$$

The map $\mathcal{J}_A$ has an inverse $\mathcal{J}_A^{-1}$ that sends a sum of distinct basis vectors in V_A to the corresponding subset of A, i.e. for $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$

$$\mathcal{J}_A^{-1}(\sum_{j=1,\ldots,k} a_{i_j}) = \{a_{i_1}, \ldots, a_{i_k}\}\,. \tag{8}$$

The following holds for every Boolean vector $v \in V_A$ and every $X \subseteq A$

$$\begin{array}{rcl} \mathcal{J}_A^{-1} \circ \mathcal{J}_A(X) & = & X \\ \mathcal{J}_A \circ \mathcal{J}_A^{-1}(v) & = & v\,. \end{array}$$

In fact, $\mathcal{J}_A$ is an isomorphism of Boolean algebras that maps subsets of A onto Boolean vectors of V_A. Indeed, $\mathcal{J}_A$ commutes with the logical connectives

$$\begin{array}{rcl} \mathcal{J}_A(X \cup Y) & = & \mathcal{J}_A(X) \vee \mathcal{J}_A(Y) \\ \mathcal{J}_A(X \cap Y) & = & \mathcal{J}_A(X) \wedge \mathcal{J}_A(Y) \text{ , for } X, Y \subseteq A \\ \mathcal{J}_A(A \setminus X) & = & \neg \mathcal{J}_A(X) \end{array}$$

Finally, for any finite set A

$$\mathcal{J}(A) = V_A\,, \tag{9}$$

for any two-sorted function $f : A \to B$ the linear map $\mathcal{J}(f) : V_A \to V_B$ is defined by

$$\mathcal{J}(f)(a)=\mathcal{J}_B(f(a)) \text{ for } a \in A\,. \tag{10}$$

The restriction of $\mathcal{J}(f) : V_A \to V_B$ to Boolean vectors is a Boolean homomorphism in $\mathcal{RI}$, because vector disjunction coincides with vector addition in this category .

Definitions (7) -(10) make sense in the category of finite dimensional Hilbert spaces as well. The maps $\mathcal{J}_A$ are Boolean isomorphisms, but the linear map $\mathcal{J}(f)$ is not a homomorphism when restricted to Boolean vectors.

The following *Embedding Lemma* is one of the reasons why $\mathcal{RI}$ is especially appropriate for natural language semantics.

Lemma 5 (Embedding Lemma)**.** *The map $\mathcal{J} : 2\mathcal{SF} \to \mathcal{RI}$ is a one-one functor that preserves the biproduct dagger compact closed structures and the logical connectives.*

The restriction of $\mathcal{J}$ to the subcategory of intrinsic maps is a functor in an arbitrary semantic category.

Proof. The proof is straight forward and essentially that given in [Preller and Sadrzadeh, 2011]. □

Only the equality $\mathcal{J}(g \circ f) = \mathcal{J}(g) \circ \mathcal{J}(f)$, which holds in $\mathcal{RI}$, does not hold in an arbitrary semantic category unless both f and g are intrinsic.

Indeed, let $f'(a) = f(a)$ if $f(a)$ is a set and $f'(a) = \{f(a)\}$ if $f(a)$ is an element. Define $g'(b)$ similarly and let $K = \bigcup_{b\in f'(a)} g'(b)$. Then on one hand,

$$(g \circ f)(a) = \bigcup_{b\in f'(a)} \bigcup_{c\in g'(b)} \{c\} = \bigcup_{c\in K} \{c\} \text{ and therefore } \mathcal{J}(g \circ f)(a) = \sum_{c\in K} c\,.$$

On the other hand, $\mathcal{J}(f)(a) = \mathcal{J}_B(f(a)) = \sum_{b\in f'(a)} b$. Hence

$$\mathcal{J}(g) \circ \mathcal{J}(f)(a) = \sum_{b\in f'(a)} \mathcal{J}(g)(b) = \sum_{b\in f'(a)} \sum_{c\in g'(b)} c = \sum_{c\in K} c\,. \tag{11}$$

The rightmost equality of (11) holds in $\mathcal{RI}$, because vector addition is idempotent. This is not the case in an arbitrary semantic category, see Example 1.

If f and g are intrinsic, however, they map elements to elements or to the empty set. Hence the sets $f'(a)$ and $g'(b)$ are either empty or singleton sets and the equalities (11) above hold. Hence again $\mathcal{J}(g) \circ \mathcal{J}(f) = \mathcal{J}(g \circ f)$.

3.3 Two-sorted truth

Let S be a fixed two-dimensional space of the semantic category $\mathcal{C}$ and $\top$ and $\bot$ its two basis vectors. Think of $\top$ as 'true' and of $\bot$ as 'false'. The Boolean vectors of S are called two-sorted *truth-values*. The basis vectors are of sort 1, the null-vector $\overrightarrow{0}$ and the full vector $\overrightarrow{1} = \top + \bot$ of sort 2. The null-vector

stands for 'neither true nor false' and $\top + \bot$ for 'partly true and partly false'. Two-sorted truth-values reflect second order properties of natural language, see Subsection 4.3 for examples.

The two-sorted connectives on S are linear maps, which are determined by their values on basis vectors. They are intrinsic and thus live in every compact closed category with biproducts.

The *two-sorted conjunction* $\mathtt{and}_S : S \otimes S \to S$, the *two-sorted disjunction* $\mathtt{or}_S : S \otimes S \to S$ and the *two-sorted negation* $\mathtt{not}_S : S \to S$ are given by

$$\begin{gathered}\mathtt{and}_S(\top \otimes \top) = \top,\ \mathtt{and}_S(\bot \otimes \top) = \mathtt{and}_S(\top \otimes \bot) = \mathtt{and}_S(\bot \otimes \bot) = \bot \\ \mathtt{or}_S(\bot \otimes \bot) = \bot,\ \mathtt{or}_S(\bot \otimes \top) = \mathtt{or}_S(\top \otimes \bot) = \mathtt{or}_S(\top \otimes \top) = \top \\ \mathtt{not}_S(\top) = \bot,\ \mathtt{not}_S(\bot) = \top\,.\end{gathered}$$

The two-sorted connectives, when restricted to basis vectors a, b of S satisfy

$$\begin{gathered}\mathtt{not}_S \circ \mathtt{not}_S \circ a = a \\ \mathtt{or}_S \circ (\mathtt{not}_S \otimes \mathtt{not}_S) \circ (a \otimes b) = \mathtt{not}_S \circ \mathtt{and}_S \circ (a \otimes b)\,.\end{gathered} \tag{12}$$

Two-sorted negation coincides with vector negation on basis vectors, but they are not identical. In fact, $\mathtt{not}_S$ is the symmetry isomorphism that exchanges the two basis vectors of S, whereas vector negation is not even an endomorphism of S

$$\begin{gathered}\mathtt{not}_S(\top) = \bot = \neg\top,\ \mathtt{not}_S(\bot) = \top = \neg\bot \\ \mathtt{not}_S(\overrightarrow{0}) = \overrightarrow{0},\ \text{whereas } \neg\overrightarrow{0} = \overrightarrow{1}\,.\end{gathered}$$

In general, the two-sorted connectives differ from the vector connectives, even on basis vectors. For example

$$\mathtt{and}_S(\bot \otimes \top) = \bot,\ \text{whereas } \bot \wedge \top = \overrightarrow{0}\,.$$

Natural language chooses the two-sorted connectives on S, a fact acknowledged by the notation.

3.4 Two-sorted predicates

Let E be any object of $\mathcal{C}$, think of the basis vectors of E as 'individuals'. Abbreviate the n-fold tensor product of E by $E^n = E \otimes \ldots \otimes E$ with n factors.

Definition 4 (**Two-sorted predicate**). *A* two-sorted predicate *is an intrinsic morphism with codomain S.*

A two-sorted predicate on E *is a two-sorted predicate that maps basis vectors of E to basis vectors of S.*

An n-ary two-sorted predicate on E *is a* two-sorted predicate on E^n.

Lemma 6. *The n-ary predicates on E together with the two-sorted connectives form a Boolean algebra.*

More precisely, assume that $p : E^n \to S$ *and* $r : E^n \to S$ *are* n*-ary two-sorted predicates on* E*. Then the linear maps*

$$\mathtt{not}_S \circ p,\ \mathtt{and}_S \circ (p \otimes r),\ \mathtt{or}_S \circ (p \otimes r),\ \mathtt{and}_S \circ (p \otimes r) \circ 2_{E^n},\ \mathtt{or}_S \circ (p \otimes r) \circ 2_{E^n}$$

are again two-sorted predicates on E *such that*

$$\begin{array}{l} \mathtt{not}_S \circ \mathtt{not}_S \circ p = p \\ \mathtt{not}_S \circ \mathtt{and}_S \circ (p \otimes r) = \mathtt{or}_S \circ ((\mathtt{not}_S \circ p) \otimes (\mathtt{not}_S \circ r)) \\ \mathtt{not}_S \circ \mathtt{and}_S \circ (p \otimes r) \circ 2_{E^n} = \mathtt{or}_S \circ ((\mathtt{not}_S \circ p) \otimes (\mathtt{not}_S \circ r)) \circ 2_{E^n}\,. \end{array} \tag{13}$$

Proof. It suffices to check the equalities (13) for basis vectors, which follows from (12). □

Two-sorted predicates coincide with one-sorted predicates on individuals. For sets they may take four different values. The following lemma tells us when.

Lemma 7. [Fundamental Property]
Let $p : V_B \to S$ *be a two-sorted predicate on* V_B *and* $Y \subseteq B$ *a subset of basis vectors. The following equivalences hold*

$$\begin{array}{l} p(\sum_{x \in Y} x) = \overrightarrow{0} \Leftrightarrow Y = \emptyset \\ p(\sum_{x \in Y} x) = \top \Leftrightarrow \forall_x (x \in Y \Rightarrow p(x) = \top) \text{ and } Y \neq \emptyset \\ p(\sum_{x \in Y} x) = \bot \Leftrightarrow \forall_x (x \in Y \Rightarrow p(x) = \bot) \text{ and } Y \neq \emptyset \\ p(\sum_{x \in Y} x) = \overrightarrow{1} \Leftrightarrow \exists_{x \in Y} \exists_{y \in Y} (p(x) = \top \text{ and } p(y) = \bot)\,. \end{array} \tag{14}$$

Proof. By linearity, $p(\sum_{x \in Y} x) = \sum_{x \in Y} p(x)$. For the last equivalence, use $\overrightarrow{1} = \top + \bot$. □

Corollary 1. *For any element* x

$$p(x) = \bot \Leftrightarrow p(x) \neq \top \Leftrightarrow \mathtt{not}_S(p(x)) = \top\,.$$

For any non-empty subset Y *of* B

$$p(Y) = \bot \Leftrightarrow \mathtt{not}_S(p(Y)) = \top\,. \tag{15}$$

In general, however, $p(Y) \neq \top$ *does not imply* $p(Y) = \bot$.

Proof. The equivalence $p(Y) = \bot \Leftrightarrow \mathtt{not}_S(p(Y)) = \top$ follows from the definition of the two-sorted negation.

By the fundamental property, $p(Y) = \bot$ is equivalent to $\forall_x (x \in Y \Rightarrow p(x) = \bot)$ for a non-empty set Y. Now assume that Y has two distinct elements x and y and that $p(x) = \top$ and $p(y) = \bot$. Then $p(Y) = \{\top, \bot\} = \overrightarrow{1} \neq \bot$. □

In the context of natural language, use different font shapes to distinguish the set $X \subseteq B$ from the vector $\mathcal{J}_B(X)$ by using italic for the former and typewriter for the latter. For example, if $Bank \subseteq B$

$$\mathtt{bank} = \sum_{x \in Bank} x = \mathcal{J}_B(Bank)\,.$$

The same applies when distinguishing a one-sorted predicate on B and the corresponding two-sorted predicate on V_B. For example, the one-sorted predicate *Rich* corresponding to $\mathtt{rich} : V_B \to S$ satisfies

$$x \in \mathit{Rich} \Leftrightarrow \mathit{Rich}(x) \Leftrightarrow \mathtt{rich}(x) = \top, \text{ for all } x \in B\,.$$

Here are a few examples how two-sorted predicates work in an arbitrary semantic category.

Note: the individuals designated by a noun form a non-empty set.

Example 2. *The following are equivalent*

$$\begin{gathered}\mathtt{rich}(\mathtt{bank}) = \top \\ \forall x(x \in \mathit{Bank} \Rightarrow \mathit{Rich}(x))\,.\end{gathered}$$

Proof. The Fundamental Property (7) implies $\mathtt{rich}(\mathtt{bank}) = \top$ if and only if $\forall x(x \in \mathit{Bank} \Rightarrow \mathtt{rich}(x) = \top)$. The latter is equivalent to $\forall x(x \in \mathit{Bank} \Rightarrow \mathit{Rich}(x))$ by definition of *Rich*. □

Example 3. *The following are equivalent*

$$\begin{gathered}\mathtt{not}_S(\mathtt{rich}(\mathtt{bank})) = \top \\ \mathtt{rich}(\mathtt{bank}) = \bot \\ \forall x(x \in \mathit{Bank} \;\Rightarrow\; x \notin \mathit{Rich})\,.\end{gathered}$$

Proof. The equivalence of the first two equalities is a special case of (15). The second equality equality is equivalent to $\forall x(x \in \mathit{Bank} \Rightarrow \mathtt{rich}(x) = \bot)$ by the Fundamental Property. Hence to $\forall x(x \in \mathit{Bank} \;\Rightarrow\; x \notin \mathit{Rich})$. □

Important: *The first order formula in Example 3 is not the negation of the first order formula in Example 2. But then, the equality* $\mathtt{rich}(\mathtt{bank}) = \bot$ *is not the negation of the equality* $\mathtt{rich}(\mathtt{bank}) = \top$*, because there are more than two truth values.*

4 Semantics via pregroup grammars

Let E be a finite dimensional space with basis B. Call its basis vectors *individuals*. Basis vectors of E correspond to singulars and sums of several basis vectors to plurals of natural language. Examples below concern unary predicates only. The generalization to ordered pairs, triples etc. of individuals is straight forward. The space S is the two-dimensional space of two-sorted truth introduced in 3.3.

4.1 The computation of meanings

Like every other categorial grammar, a pregroup grammar is given by a lexicon and a logical calculus. Syntactical analysis consists in a proof in the logical calculus. The pregroup calculus is *compact bilinear logic* where proofs identify with morphisms in the free compact bicategory $\mathcal{C}2(\mathcal{B})$ generated by a partially ordered set $\mathcal{B}$.

The 1-cells are called *types* and the tensor product is concatenation. Hence, a *type* is a string of simple types, where a *simple type* is either an element of $x, y, \dots \in \mathcal{B}$ or an iterated adjoint $x^\ell, y^\ell, \dots, x^{\ell\ell}, y^{\ell\ell}, \dots, x^r, y^r, \dots, x^{rr}, y^{rr}, \dots$ etc. The types $x, y, \dots \in \mathcal{B}$ are called *basic types*. They stand for grammatical notions.

As a consequence, every functor from $\mathcal{B}$ into a semantic category $\mathcal{C}$ extends into a functor from $\mathcal{C}2(\mathcal{B})$ to $\mathcal{C}$ preserving the structure of compact bicategories.

A *lexicon* is a finite list of entries. An *entry* is a triple $w : T :: m$, where w is a word, T a type and m a meaning expression.

This description differs from the original one in [Lambek, 1999]. There, only pregroup 'dictionaries' are considered where the entries are pairs $w : T$, without meaning expressions. The latter must be added explicitly, because the functional semantics that the higher order types confer to categorial grammars has been lost by the pregroup types.

The *meaning* in the entry is a formal expression $m : I \to V$ in the language of compact closed categories or, equivalently, a string of two-sorted functions. It depends functionally on the word and the type in the entry.

Consider the following entries

$$
\begin{array}{ll}
\textit{no} : \boldsymbol{s}\boldsymbol{s}^\ell \boldsymbol{n}_2 \boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{no}}} S \otimes S^* \otimes E \otimes E^* & \textit{some} : \boldsymbol{n}_2 \boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{some}}} E \otimes E^* \\
\textit{are} : \boldsymbol{n}_2{}^r \boldsymbol{s}\boldsymbol{s}^\ell \bar{\boldsymbol{n}} :: I \xrightarrow{\overline{\texttt{are}}} E^* \otimes S \otimes S^* \otimes E & \textit{big} : \boldsymbol{c}_2 \boldsymbol{c}_2{}^\ell :: I \xrightarrow{\overline{\texttt{big}}} E \otimes E^* \\
\textit{and} : \boldsymbol{s}^r \boldsymbol{s}\boldsymbol{s}^\ell :: I \xrightarrow{\overline{\texttt{ands}}} S^* \otimes S \otimes S^* & \textit{banks} : \boldsymbol{c}_2 :: I \xrightarrow{\overline{\texttt{bank}}} E \\
 & \textit{rich} : \bar{\boldsymbol{n}}^r \boldsymbol{s} :: I \xrightarrow{\overline{\texttt{rich}}} E^* \otimes S
\end{array}
$$

The basic types $\boldsymbol{c}_2, \boldsymbol{n}_2, \bar{\boldsymbol{n}}, \boldsymbol{s}$, stand for 'plural count noun', 'plural noun phrase', 'dummy noun phrase','sentence' in that order. Moreover, $\boldsymbol{c}_2 < \boldsymbol{n}_2$.

The properties of the meaning vector m in $w : T :: m$ depend on the logical content of the word, given in due course, and on the type.

The lexicon defines a *canonical* functor from $\mathcal{B}$ into the compact closed category $\mathcal{C}$. For example, in the list of entries above, the basic type $\boldsymbol{s}$ is interpreted by S and each of the basic types $\boldsymbol{c}_2, \boldsymbol{n}_2, \bar{\boldsymbol{n}}$ by E . The canonical functor maps the inequality $\boldsymbol{c}_2 < \boldsymbol{n}_2$ to the identity 1_E and left and right adjoints to 'the' adjoint space, because right and left adjoints may be identified in a symmetric monoidal category.

All meanings are names of morphisms, up to a symmetry isomorphism that arranges the factors of the tensor product in the order given by the simple types. For example, $\overline{\texttt{big}} = \sigma_{E^*,E} \circ \ulcorner \textit{big} \urcorner : 1 \to E \otimes E^*$ where $\texttt{big} : E \to E$.

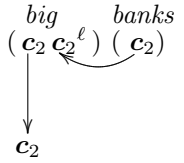
The meaning of grammatical strings involves both the meaning vectors of the words and a syntactical analysis of the string by the grammar.

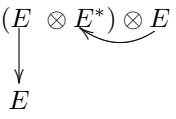
A string of words $w_1 \dots w_n$ is *grammatical* if there are entries $w_1 : T_1 :: m_1, \dots,\ w_n : T_n :: m_n$ and a basic type $\boldsymbol{b}$ such that

$$T_1 \dots T_n \vdash \boldsymbol{b}$$

is provable in compact bilinear logic. Otherwise said, if there is a morphism $f : T_1 \dots T_n \to \boldsymbol{b}$ in the syntactic category. Due to a theorem in [Lambek, 1999] the graph of the proof involves only underlinks and is called a *reduction*.

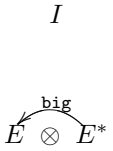
For example, the reduction corresponding to the graph on the left below analyses *big banks* as a noun-phrase. The graph on the right is the corresponding morphism in the semantic category.

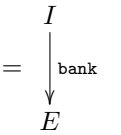
$1_{\boldsymbol{c}_2} \otimes \epsilon_{\boldsymbol{c}_2} =$ 

$r_1 = 1_E \otimes \epsilon_E =$ 

The reason why the syntactic category may not be symmetric is obvious in this example. The type $\boldsymbol{c}_2\boldsymbol{c}_2\boldsymbol{c}_2{}^{\ell}$ of the non-grammatical string *banks big* has no reduction to a basic type. If we had symmetry all order variants of a grammatical string would be grammatical.

The meaning vector of a lexical entry also identifies with a graph, for example

$I \xrightarrow{\overline{\texttt{big}}} E \otimes E^* =$ 

$\overline{\texttt{bank}} : I \to E =$ 

.

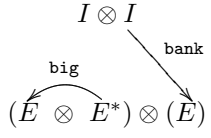
Again, the domain of the morphism is at the top of the graph, the codomain at the bottom.

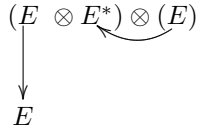
For a grammatical string $w_1 \dots w_n$, there is a choice of entries $w_1 : T_1 :: m_1, \dots, w_n : T_n :: m_n$ and a reduction of $T_1 \dots T_n$ to a basic type. The corresponding *meaning* is

$$r \circ (m_1 \otimes \dots \otimes m_n)\,,$$

where the linear map r is obtained by applying the canonical functor to the reduction. Hence, the meaning of a string involves both the tensor product of the words and a reduction.

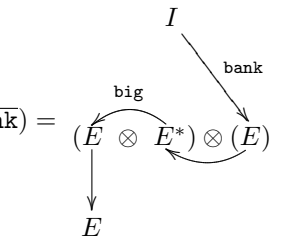
The meaning of a string can be computed graphically. Connect the graphs at there joint interface and follow the paths from top to bottom picking up the labels along the way. For example, the graphs

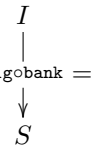
$\overline{\texttt{big}} \otimes \overline{\texttt{bank}} =$ 

$r_1 =$ 

when connected at there joint interface compute to

$$r_1 \circ (\overline{\texttt{big}} \otimes \overline{\texttt{bank}}) = \quad = \texttt{big} \circ \texttt{bank} = \texttt{big} \circ \texttt{bank}\,.$$



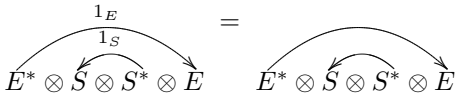


The meaning vector $\overline{\texttt{are}} : I \to E^* \otimes S \otimes S^* \otimes E$ is up to a symmetry isomorphism the name of the linear map $\texttt{are} : E \otimes S \to E \otimes S$.
The *logical property* of the word *are* is rendered by

$$\texttt{are} = 1_E \otimes 1_S : E \otimes S \to E \otimes S\,.$$

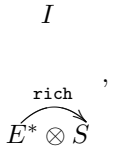
Hence of the graph of $\overline{\texttt{are}}$ is

$$\overline{\texttt{are}} = \quad = \quad .$$



The meaning vector $\overline{\texttt{rich}} : I \to E^* \otimes S$ is represented by the graph

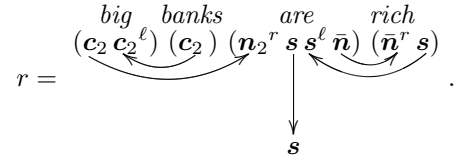
$$\overline{\texttt{rich}} = \quad ,$$



where $\texttt{rich} : E \to S$ is a unary two-sorted predicate.

The reduction of the sentence *big banks are rich* is the graph

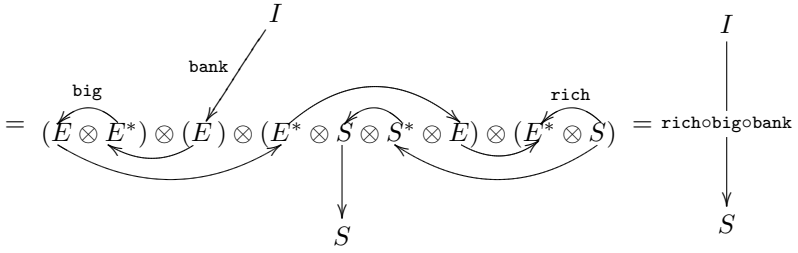
$$r = \quad .$$



Compute the meaning by composing the tensor product of the word vectors

with the reduction

$r \circ (\overline{\texttt{big}} \otimes \overline{\texttt{bank}} \otimes \overline{\texttt{are}} \otimes \overline{\texttt{rich}})$

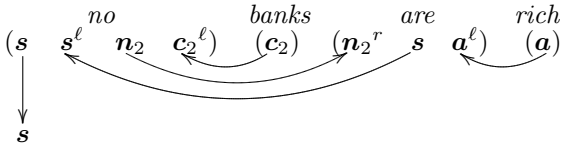


$= \texttt{rich} \circ \texttt{big} \circ \texttt{bank}\,.$

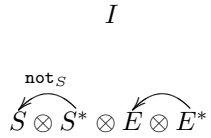
The sentence *All banks are rich* is computed by the same graph except that the label `big` is replaced by the label `all`. A similar remark applies to the sentence *Some banks are rich*.

The last example concerns the computation of the meaning vector of the sentence *no banks are rich*.

- the reduction of the sentence is

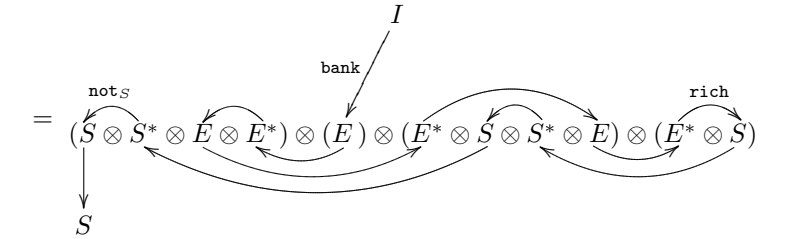


- the meaning vector $\overline{\texttt{no}} : I \to S \otimes S^* \otimes E \otimes E^*$ is defined as the matrix of $\texttt{not}_S \otimes 1_E : S \otimes E \to S \otimes E$ with the corresponding graph



- the meaning of the sentence *no banks are rich* is

$r \circ (\overline{\texttt{no}} \otimes \overline{\texttt{bank}} \otimes \overline{\texttt{are}} \otimes \overline{\texttt{rich}})$



$= \texttt{not}_S \circ \texttt{rich} \circ \texttt{bank}\,.$

'Walking graphs' makes the computation linear in the number of links. The number of links is proportional to the length of the string of words, because the lexicon is finite and thus the length of its types is bounded.

4.2 The logical content of words

The preceding examples mention the logical content of some words. More generally, the description of the logical content can be organized according to the type of the words. We postulate

1. *Any* $f : E^n \to S$ *occurring in a lexical entry is an* n*-ary two-sorted predicate.*
2. *Any* $f : E \to E$ *occurring in a lexical entry is an intrinsic projector.*
3. *Any* $f : I \to E$ *occurring in a lexical entry is a Boolean vector.*

For example, adjectives in predicative position and intransitive verbs are unary two-sorted predicates. Adjectives in attributive position and determiners like *the*, *some*, *all*, have lexical entries of the form $det : \boldsymbol{n}_i \boldsymbol{c}_i^\ell :: \overline{\texttt{det}} : I \to E \otimes E^*$, with a corresponding meaning map $\texttt{det} : E \to E$. By the postulate above, they are intrinsic projectors. The universal determiner satisfies an even stronger property

$$\texttt{all} = 1_E \,.$$

Assume $\texttt{word} : E \to E$ is an intrinsic projector occurring in the lexicon. Use the abbreviation

$$\mathit{Word} = \texttt{word}(B) \,.$$

This abbreviation, combined with Equality (6) implies

$$\texttt{word}(Y) = \mathit{Word} \cap Y \,. \tag{16}$$

For example, $\mathit{Big} = \texttt{big}(B)$ and $\texttt{big}(\mathit{Bank}) = \mathit{Big} \cap \mathit{Bank}$

4.3 Examples

The examples below concern sentences and their meaning vectors. Represent the truth of a sentence by the fact that the meaning vector computes to $\top$. Then show that this representation coincides with the 'usual' translation of the sentence in logic.

Example 4. *All banks are rich* / $\texttt{rich}(\texttt{all}(\texttt{bank}))$

The following are equivalent

$$\forall x(x \in \mathit{Bank} \Rightarrow \mathit{Rich}(x))$$
$$\texttt{rich}(\texttt{all}(\texttt{bank})) = \top \,.$$

Proof. Recall that $\texttt{all} = 1_E$. Hence $\texttt{rich}(\texttt{all}(\texttt{bank})) = \texttt{rich}(\texttt{bank})$. For the proof of the equivalence of $\texttt{rich}(\texttt{bank}) = \top$ with $\forall x(x \in \mathit{Bank} \Rightarrow \mathit{Rich}(x))$ confer to Example 2. □

Example 5. *Big banks are rich* / $\texttt{rich}(\texttt{big}(\texttt{bank}))$

The following are equivalent

$$\forall x(x \in \mathit{Big} \cap \mathit{Bank} \Rightarrow \mathit{Rich}(x))$$
$$\texttt{rich}(\texttt{big}(\texttt{bank})) = \top \,.$$

Proof. Recall that $\texttt{bank} = \sum_{x \in Bank} x$ and therefore the vector $\texttt{big}(\texttt{bank})$ is identified with the set $\texttt{big}(Bank) = Big \cap Bank$ by (16). The equivalence now follows from the Fundamental Property (7). □

Example 6. *No banks are rich* / $\texttt{not}_S(\texttt{rich}(\texttt{bank}))$

The following are equivalent

$$\texttt{not}_S(\texttt{rich}(\texttt{bank})) = \top$$
$$\forall x(x \in Bank \;\Rightarrow\; x \notin Rich)$$

Proof. See Example 3. □

Example 7. *Some banks are rich*/$\texttt{rich}(\texttt{some}(\texttt{bank}))$

The implication

$$\texttt{rich}(\texttt{some}(\texttt{bank})) = \top \Rightarrow \exists_x(x \in Bank \,\&\, Rich(x))$$

holds, but its converse does not hold in general.

Proof. The equality $\texttt{rich}(\texttt{some}(\texttt{bank})) = \top$ implies $\texttt{some}(\texttt{bank}) \neq \emptyset$ and $\forall_x(x \in \texttt{some}(\texttt{bank}) \Rightarrow \texttt{rich}(x) = \top)$, by the Fundamental Property. The first order formula follows, because $\texttt{some}(\texttt{bank}) \subseteq \texttt{bank}$.

If the first order formula holds, take a witness $b \in Bank$ for which *Rich(b)* holds. Let $\texttt{some}_b$ be the intrinsic projector that maps b to itself and every other individual to the null vector. Clearly $\texttt{some}_b(\texttt{bank}) = b$ and $\texttt{rich}(b) = \top$. Hence,

$$\texttt{rich}(\texttt{some}_b(\texttt{bank})) = \top,$$

but this does not imply that the particular intrinsic projector $\texttt{some}_b$ coincides with the original $\texttt{some}$. □

Natural language confronts us with a problem. On one hand, the interpretation of '*some Y*' changes from occurrence to occurrence, like in *Some banks are rich and some banks are not rich.* On the other hand, '*some Y*' acts like a name for a well determined set. In *Some banks are rich. They scare me*, the personal pronoun *they* refers to the set *some banks* of the preceding sentence.

The interpretation of $\texttt{some}$ given above may vary from occurrence to occurrence and in the same time it defines a set at each occurrence, which is available for later reference, e.g. $\texttt{they} = \texttt{some}(\texttt{bank})$.

The interpretation by a generalized quantifier, see [Barwise and Cooper, 2002], takes into account the change of meaning in different occurrences, but it does not construct the set to which the noun phrase refers.

5 Compositional semantics in concept spaces

Quantum logic stands for 'logic of projectors in a concept space' and *concept* for 'Boolean vector in a concept space'.

5.1 Classical propositional calculus in concept spaces

Let $P = \{p_1, \ldots, p_d\}$ be a non-empty set. Call *compound system* or *concept space* the tensor product

$$C(P) = C(p_1) \otimes \ldots \otimes C(p_d)\,,$$

where $C(p_i)$ is a 2-dimensional space with basis vectors $p_{i\top}, p_{i\bot}$, for $i = 1, \ldots, d$.

The space $C(p_i)$ is a 'basic variable' in quantum protocols and a 'basic concept' in semantics for natural language. For example, key-words of Roget's (or the speaker's mental) thesaurus provide sets of basic concepts.

Any basis vector b_f of $C(P)$ is a tensor product of basis vectors of the factors

$$b_f = f(1) \otimes \ldots \otimes f(d), \text{ where } f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\}\ .$$

Due to the fine-grained structure of the basis vectors, the Boolean algebra of intrinsic projectors of a concept space is isomorphic to the Boolean algebra freely generated by the set P. The rest of this subsection is devoted to the proof of this fact.

For every $i = 1, \ldots, d$, define the two so-called *elementary vectors*

$$\begin{array}{ll} \overrightarrow{p_i} & = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes p_{i\top} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \sum_{f, f(i)=p_{i\top}} b_f \\ \overrightarrow{\neg p_i} & = \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} \otimes p_{i\bot} \otimes \overrightarrow{1} \otimes \ldots \otimes \overrightarrow{1} = \sum_{f, f(i)=p_{i\bot}} b_f\,. \end{array}$$

The two elementary vectors defined by $p_i \in P$ are orthogonal to each other. In fact, one is the negation of the other

$$\neg\overrightarrow{p_i} = \overrightarrow{\neg p_i} \ \text{ and } \ \neg(\overrightarrow{\neg p_i}) = \overrightarrow{p_i}\,.$$

Every Boolean vector can be written as a disjunction of conjunctions of elementary vectors. Indeed, let $\{j_1, \ldots, j_k\}$ be a subset of $\{1, \ldots, d\}$. Assume that $g \in \prod_{i=1}^{d} \left\{p_{i\top}, p_{i\bot}, \overrightarrow{1}\right\}$ satisfies for $i = 1, \ldots, d$

$$g(i) \in \{p_{i\top}, p_{i\bot}\} \ \text{ if and only if } i \in \{j_1, \ldots, j_k\}\ . \tag{17}$$

The *partial choice vector* associated to g is

$$v_g = g(1) \otimes \ldots \otimes g(d)\,.$$

Lemma 8. *Every partial choice vector v_g is a conjunction of elementary vectors. In particular, every basis vector is a conjunction of elementary vectors.*

Proof. Assume that g satisfies(17). Let $q_{j_l} = g(j_l) \in \{p_{j_l\top}, p_{j_l\bot}\}$, $l = 1, \ldots, k$, and $G = \left\{f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\} \ : \ f(j_l) = q_{j_l}, \text{ for } l = 1, \ldots, k\right\}$. Then

$$v_g = \sum_{f \in G} b_f = \overrightarrow{q_{j_1}} \wedge \cdots \wedge \overrightarrow{q_{j_k}}\,. \tag{18}$$

□

Theorem 2. *The free Boolean algebra generated by P is isomorphic to the lattice of intrinsic projectors of $C(P)$. The map $p \mapsto \overrightarrow{p}$ extends to an isomorphism $\mathcal{K}$ from $B(P)$ onto the lattice of Boolean vectors of $C(P)$.*

Proof. Partial choice vectors are Boolean vectors by (18). Hence, the map $p \mapsto \overrightarrow{p}$ extends to a unique Boolean homomorphism $\mathcal{K}$ from $B(P)$ into the Boolean algebra of Boolean vectors of $C(P)$. A classical theorem, [**?**], states that the free Boolean algebra $B(P)$ is isomorphic to the set algebra generated by the following subsets of $\prod_{j=1}^{d} \{0,1\}$

$$p_i \simeq \left\{ h \in \prod_{j=1}^{d} \{0,1\} \ : \ h(i) = 1 \right\}, \quad \neg p_i \simeq \left\{ h \in \prod_{j=1}^{d} \{0,1\} \ : \ h(i) = 0 \right\},$$

where i varies from 1 to d. Every singleton set $\{h\}$ can be written as a conjunction of subsets of the form p_i or $\neg p_i$. Therefore the homomorphism $\mathcal{K}$ maps $\{h\}$ to the corresponding basis vector in $C(P)$. It follows that the homomorphism $\mathcal{K}$ is onto and one-one, because every Boolean vector can be written uniquely as a sum of basis vectors.

Compose $\mathcal{K}$ with the isomorphism $\mathcal{H}$ of Theorem 1 to obtain the isomorphism $\mathcal{H} \circ \mathcal{K}$ onto the lattice of intrinsic projectors. □

By Theorem 2, the *elementary projectors* $p_{\overrightarrow{p_i}}$ and $p_{\neg\overrightarrow{p_i}}$ play an important role in the lattice of projectors of $C(P)$:

1. Every intrinsic projector is a finite disjunction of finite conjunctions of elementary projectors
2. The lattice of intrinsic projectors in a compound system is the classical propositional calculus modulo equiderivability.
3. One can use *induction on the complexity of propositions* for defining and proving properties of Boolean vectors/intrinsic projectors.

Propositional complexity creates a somewhat unusual hierarchy on subspaces: The elementary subspace $E_{\overrightarrow{p_i}}$ has complexity 0 respectively $\neg E_{\overrightarrow{p_i}} = E_{\neg\overrightarrow{p_i}}$ has complexity 1, but both have dimension 2^{d-1}. The one-dimensional subspace generated by a single basis vector b_f has complexity $d-1$ if $f(i) = p_i$ for $i = 1, \ldots, d$ and complexity d otherwise.

5.2 Concept spaces and two-sorted truth

A *classification system* consists of

1. a set B (of individuals, pairs of individuals etc.)
2. a set $P = \{p_1, \ldots, p_d\}$ (of properties)
3. a relation $\models \subseteq B \times P$

Read $x \models p$ as 'x satisfies p'.

Extend the relation $\models$ to arbitrary concepts in $C(P) = C(P_1) \otimes \ldots \otimes C(P_d)$ for every individual $x \in B$ using induction on the complexity of concepts

$$\begin{array}{ll} x \models \overrightarrow{p_i} & \text{if and only if } \; x \models p_i \\ x \models \neg v & \text{if and only if } \; x \not\models v \\ x \models v \wedge w & \text{if and only if } \; x \models v \;\&\; x \models w \\ x \models v \vee w & \text{if and only if } \; \models v \text{ or } x \models w \,. \end{array}$$

Clearly, either $x \models v$ or $x \models \neg v$ holds for every individual $x \in B$ and every concept $v \in C(P)$.

Extend satisfaction to every non-empty subset Y of B and every concept v

$$Y \models v \text{ if and only if } x \models v \text{ for all } x \in Y \,. \tag{19}$$

Read $Y \models v$ as 'Y has property v in general'.

Note that $Y \not\models v$ and $Y \not\models \neg v$ may hold simultaneously. It suffices that Y has an element satisfying v and another one that does not satisfy v.

The satisfaction system induces a representation of (sets of) individuals by concepts in $C(P)$. For any $x \in B$ let

$$\begin{array}{l} q(x)_i = p_{i\top} \text{ if } x \models p_i \\ q(x)_i = p_{i\bot} \text{ if } x \not\models p_i \,, \end{array}$$

for $i = 1, \ldots, d$. This choice determines a basis vector, the *concept internalizing* x,

$$v_x = q(x)_1 \otimes \ldots \otimes q(x)_d \,.$$

For any non-empty subset $Y \subseteq B$ define the *concept internalizing* Y

$$v_Y = \sum_{x \in Y} v_x \,.$$

Different individuals may be internalized by the same basis vector. This means that they are indiscernible by the properties listed in P.

Lemma 9. *For any concept $c \in C(P)$ and any individual $x \in B$*

$$x \models c \text{ if and only if } \; v_x \leq c \,. \tag{20}$$

In particular, for any basis vector $b_f \in C(P)$

$$x \models b_f \text{ if and only if } v_x = b_f \,. \tag{21}$$

For $Y \neq \emptyset$

$$Y \models c \text{ if and only if } \; v_Y \leq c \,. \tag{22}$$

Proof. Show (20), the equivalence concerning individuals, by induction on the propositional complexity of c. Equivalence (21) is a particular case of (20).

The equivalence concerning sets, (22), now follows from the equivalence for individuals. □

One consequence of the lemma above is that satisfaction in a classification system coincides with the conditional logic for Boolean vectors/projectors. Indeed, the inequality $v \leq w$ is equivalent to $v \to w = \overrightarrow{1}$, where the full vector $\overrightarrow{1}$ stands for 'true'.

Another consequence is that the concept v_x is the best possible description of the individual x in the classification system and the same holds for v_Y and the set Y.

5.3 Intrinsic projectors and two-sorted predicates

Let E be a space with basis B and $(B, P = \{p_1, \ldots, p_d\}, \models)$ a satisfaction system. For any $p \in P$, define a two sorted predicate $p(\,.\,) : E \to S$ by the condition

$$p(x) = \begin{cases} \top & \text{if } x \models p \\ \bot & \text{if } x \not\models p \end{cases}, \text{ for all } x \in B\,.$$

Conversely, given a set of two-sorted predicates $P = p_1, \ldots, p_d$ on E, define a satisfaction relation $\models \subseteq B \times P$ such that

$$x \models p \text{ if and only if } p(x) = \top \text{ , for all } x \in B,\, p \in P\,.$$

Recall that $x \models v$ is equivalent to $v_x \leq v$. The expressiveness of the logic remains unchanged if the individuals are replaced by the basis vectors internalizing them. Indeed, individuals x, y for which $v_x = v_y$ are indiscernible in the logic.

The compound system $C(P)$ is endowed with a *canonical satisfaction system*, namely

$$\begin{array}{rcl} P & = & \{p_1, \ldots, p_d\} \\ B & = & \left\{ b_f \,:\, f \in \prod_{i=1}^{d} \{p_i\top, p_i\bot\} \right\} \\ x \models p & \Leftrightarrow & x \leq \overrightarrow{p},\ \text{for all } x \in B,\, p \in P\,. \end{array}$$

In the canonical satisfaction system, a basis vector can be both an individual and a concept. More generally, every Boolean vector is both a set of individuals and a concept. For example, $\mathtt{bank} = \sum_{x \in Bank} x = \mathcal{J}_B(Bank)$ and $v_{Bank} = \mathtt{bank}$.

Define a map $\overrightarrow{p} \mapsto \overrightarrow{p}(\,.\,)$ from the elementary vectors $\overrightarrow{p} \in C(P)$ to two-sorted predicates on $C(P)$ by stipulating

$$\begin{array}{l} \overrightarrow{p}(x) = \top \quad \Leftrightarrow\ x \leq \overrightarrow{p} \\ \overrightarrow{p}(x) = \bot \quad \Leftrightarrow\ x \not\leq \overrightarrow{p} \end{array} \text{ , for all } x \in B\,. \tag{23}$$

Theorem 3. *The map $\overrightarrow{p} \mapsto p(\,.\,)$ extends to a Boolean isomorphism $\mathcal{L}$ from the Boolean vectors of $C(P)$ onto the Boolean algebra of two-sorted predicates on $C(P)$ satisfying for $\mathcal{L}(v) = v(\,.\,)$*

$$\begin{array}{ll} (\mathtt{not}_S \circ v)(\,.\,) & = (\neg v)(\,.\,) \\ \mathtt{and}_S \circ (v(\,.\,) \otimes w(\,.\,)) \circ 2_E & = (v \wedge w)(\,.\,)\,. \end{array} \tag{24}$$

Moreover, the following equivalences hold if $w \neq \overrightarrow{0}$

$$\begin{array}{lll} v(w) = \top & \Leftrightarrow & w \leq v \\ v(w) = \bot & \Leftrightarrow & w \leq \neg v\,. \end{array} \tag{25}$$

Proof. The existence of the Boolean homomorphism $v \mapsto v(\,.\,)$ satisfying (24) is guaranteed by Theorem 2 .

Start the proof of (25) by showing first that

$$\begin{array}{lll} v(x) = \top & \Leftrightarrow & x \leq v \\ v(x) = \bot & \Leftrightarrow & x \leq \neg v \end{array} \text{ , for all } x \in B\,. \tag{26}$$

Use induction on the propositional complexity of v . If the complexity is 0 $v = \overrightarrow{p}$. The two equivalences of (26) hold for $v = \overrightarrow{p}$ by the stipulations (23) and the fact that $x \leq \neg\overrightarrow{p}$ if and only if $x \not\leq \overrightarrow{p}$.

For the induction step, assume that (25) holds for v . Recall that $\mathtt{not}_S$ is the symmetry isomorphism that exchanges the two basis vectors $\top$ and $\bot$. Thus

$$\top = (\neg v)(x) = (\mathtt{not}_S \circ v)(x) \Leftrightarrow v(x) = \bot \Leftrightarrow x \leq \neg v \quad ,$$

because $v(x) = \bot \Leftrightarrow\ x \leq \neg v$ by assumption.

Next, assume that (26) holds for the concepts v and w . Then

$$(v \wedge w)(x) = (\mathtt{and}_S \circ (v \otimes w) \circ 2_E)(x) = \mathtt{and}_S \circ (v(x) \otimes w(x))\,.$$

Therefore, assuming $(v \wedge w)(x) = \top$ is the same as assuming $v(x) = \top$ and $w(x) = \top$, according to the definition of $\mathtt{and}_S$. The latter two equalities are equivalent to $x \leq v$ and $x \leq w$ by induction hypothesis. Hence $(v \wedge w)(x) = \top$ is equivalent to $x \leq v \wedge w$. The proof of the equivalence $(v \wedge w)(x) = \bot \Leftrightarrow\ x \leq \neg(v \wedge w)$ is similar. This concludes the proof by induction of (26).

Let $w \neq \overrightarrow{0}$ be an arbitrary Boolean vector. Then $v(w) = \top$ if and only if $x \leq w$ implies $v(x) = \top$ for all $x \in B$ if and only if $x \leq w$ implies $x \leq v$ for all $x \in B$ if and only if $w \leq v$. Thus (25) holds.

Next, (26) implies that the homorphism $v \mapsto v(\,.\,)$ is one-one. Finally, for the proof that is also onto, let $r : C(P) \to S$ be any two-sorted predicate on $C(P)$. Let $K = \{x \in B : r(x) = \top\}$ and $v = \sum_{x \in K} x$. Recall that a two-sorted predicate maps basis vectors to basis vectors. Hence, $r(x) = \bot$ for all basis vectors $x \notin K$. Thus $r(x) = \top$ if and only if $x \leq v$ and $r(x) = \bot$ if and only if $x \leq \sum_{x \in B \setminus K} x = \neg v$. Therefore $r = v(\,.\,)$ by (25). □

This theorem has a generalization to an arbitrary space of individuals. The homomorphism $\mathcal{L}$ is defined and satisfies (24) - (26). It is still one-one, but it is not necessarily onto.

Note that the homomorphism $\mathcal{L}$ maps the full vector $\overrightarrow{1} \in C(P)$ to the predicate that maps every basis vector to $\top$ and the null vector $\overrightarrow{0}$ to the predicate that takes value $\bot$ for all basis vectors in B .

Switching from Boolean vectors to intrinsic projectors, identify the intrinsic projector p_v with the two-sorted predicate $v(\,.\,)$ via the isomorphism $\mathcal{L} \circ \mathcal{H}^{-1}$. The equalities (24) and (25) become for all Boolean vectors $v, w \in C(P)$

$$(\mathtt{not}_S \circ v)(\,.\,) = p_v^{\perp}, \quad \mathtt{and}_S \circ (v(\,.\,) \otimes w(\,.\,)) \circ 2_{C(P)} = p_v \circ p_w$$

$$\begin{array}{l} v(w) = \top \Leftrightarrow p_v(w) = w \\ v(w) = \perp \Leftrightarrow p_v(w) = w \\ v(w) = \perp \Leftrightarrow p_v^{\perp}(w) = \overrightarrow{0} \end{array} \text{ , for } w \neq \overrightarrow{0}\,.$$

The third possible value $\top + \perp$ of two-sorted predicates leads to the following characterization in terms of projectors.

Lemma 10. *Let $v(\,.\,)$ be a two-sorted predicate and p_v the corresponding projector. Then for any Boolean vector w*

$$v(w) = \top + \perp \Leftrightarrow (p_v(w) \neq \overrightarrow{0}\ \&\ p_v^{\perp}(w) \neq \overrightarrow{0})\,.$$

Proof. Let W be the set of basis vectors for which $w = \sum_{z \in W} z$. Assume the lefthand equality. By the Fundamental Property (7), this means that the two-sorted predicate v takes some $x \in W$ to $\top$ and some $y \in W$ to $\perp$. Replace the equality $v(x) = \top$ by the corresponding projector equality $p_v(x) = x$. Similarly, $v(y) = \perp$ may be read as $p_v^{\perp}(y) = y$. □

Theorems 1 and 3 bring a new understanding to projectors in a compound system $C(P)$. Boolean vectors, intrinsic projectors of $C(P)$ and two-sorted predicates on $C(P)$ are interchangeable. All sentences expressible in two-sorted first order logic are expressible in quantum logic. It suffices to make $E = C(P)$ in Subsection 4.1.

The sample sentences of Section (4.3) have two interpretations in $C(P)$. One is a two-sorted predicate and the other one a projector. The former equals $\top$ exactly when the latter equals $1_{C(P)}$. Similarly, the former equals $\perp$ exactly when the latter equals $0_{C(P)}$.

All banks are rich / $\mathtt{rich}(\mathtt{all}(\mathtt{bank}))$ / $p_{\mathtt{bank}} \rightarrow\ p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{all}(\mathtt{bank})) = \top \text{ if and only if } p_{\mathtt{bank}} \rightarrow p_{\mathtt{rich}} = 1_{C(P)}\,.$$

Big banks are rich / $p_{\mathtt{big}} \circ p_{\mathtt{bank}} \rightarrow p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{big}(\mathtt{bank})) = \top \text{ if and only if } p_{\mathtt{big}} \circ p_{\mathtt{bank}} \rightarrow p_{\mathtt{rich}} = 1_{C(P)}\,.$$

No banks are rich / $\mathtt{not}_S(\mathtt{rich}(\mathtt{bank}))$ / $p_{\mathtt{bank}} \rightarrow p_{\mathtt{rich}}^{\perp}$

$$\mathtt{not}_S(\mathtt{rich}(\mathtt{bank})) = \top \text{ if and only if } p_{\mathtt{bank}} \rightarrow p_{\mathtt{rich}}^{\perp} = 1_{C(P)}\,.$$

Some banks are rich / $\mathtt{rich}(\mathtt{some}(\mathtt{bank}))$ / $p_{\mathtt{some(bank)}} \rightarrow p_{\mathtt{rich}}$

$$\mathtt{rich}(\mathtt{some}(\mathtt{bank})) = \top \text{ if and only if } p_{\mathtt{some(bank)}} \rightarrow p_{\mathtt{rich}} = 1_{C(P)}\,.$$

These equivalences concern Boolean vectors. The next subsection deals with non Boolean vectors.

5.4 States in a concept space

In this Section, $\mathcal{C}$ is the category of finite dimensional real Hilbert spaces.

A satisfaction relation requires a yes or no answer for every individual and every basic property p_i. For practical reason such a precise information may not be available. Assume that real numbers $\alpha_{iY} \in [0,1]$ are given for a set of individuals Y and $i = 1, \ldots, d$. Interpret α_{iY} as the probability that an arbitrary individual in Y has property p_i.

Let $0 \leq \alpha_{iY} \leq 1$ and $\beta_{iY} = 1 - \alpha_{iY}$ for $i = 1, \ldots, d$. The set Y is represented in $C(p_1) \otimes \ldots \otimes C(p_d)$ by its *state vector*

$$\mu_Y = (\alpha_{1Y} p_{1\top} + \beta_{1Y} p_{1\bot}) \otimes \ldots \otimes (\alpha_{dY} p_{d\top} + \beta_{dY} p_{d\bot}) .$$

Lemma 11. *The coordinates of μ_Y define a probability on the event space $B(P)$ generated by the $\overrightarrow{p_i}$'s. Moreover, α_i is equal to the sum of the coordinates of $\overrightarrow{p_i} \wedge \mu_Y$ and β_i to the sum of the coordinates of $\neg\overrightarrow{p_i} \wedge \mu_Y$.*

Proof. Let γ_f be the coordinate of μ_Y for $f \in \prod_{i=1}^{d} \{p_{i\top}, p_{i\bot}\}$. Then $\overrightarrow{p_i} \wedge \mu_Y = \sum_{f,\, f(i)=p_{d\top}} \gamma_f b_f$. Hence the assertions follow from the equalities

$$\sum_f \gamma_f = 1, \quad \alpha_i = \sum_{f(i)=p_{i\top}} \gamma_f, \quad \beta_i = \sum_{f(i)=p_{i\bot}} \gamma_f .$$

Prove the first equality by induction on d. The case $d = 1$ is trivial. For the induction step, let $d' = d - 1$, $P' = \{1, \ldots, d'\}$ and

$$\mu'_Y = (\alpha_{1Y} p_{1\top} + \beta_{1Y} p_{1\bot}) \otimes \ldots \otimes (\alpha_{d'Y} p_{d'\top} + \beta_{d'Y} p_{d'\bot}) .$$

Let δ_g be the coordinate of μ'_Y in $C(P')$, i.e. $\mu'_Y = \sum_{g \in \prod_{i=1}^{d'} \{p_{i\top}, p_{i\bot}\}} \delta_g b_g$. Then $\sum_g \delta_g = 1$ by induction hypothesis. We have

$$\mu_Y = \mu'_Y \otimes (\alpha_{dY} p_{d\top} + \beta_{dY} p_{d\bot}) = \sum_g \delta_g \alpha_{dY} (b_g \otimes p_{d\top}) + \sum_g \delta_g \beta_{dY} (b_g \otimes p_{d\bot}) .$$

This finishes the proof, because for every basis vector $b_f \in C(P)$ there is a unique $g \in \prod_{i=1}^{d-1} \{p_{i\top}, p_{i\bot}\}$ such that either $b_f = b_g \otimes p_{d\top}$ or $b_f = b_g \otimes p_{d\bot}$. $\square$

The projector $p_{\overrightarrow{p_i}}$ of $C(P)$ maps the state vector μ_Y to a vector $p_{\overrightarrow{p_i}}(\mu_Y) = \overrightarrow{p_i} \wedge \mu_Y$, the coordinates of which sum up to α_i. Hence the $p_{\overrightarrow{p_i}}$ returns the probability that an arbitrary individual in Y satisfies p_i.

Return to vector semantics in information retrieval systems. Choose a set $P = p_1, \ldots, p_d$ of basic properties, for example the most frequent words in a (set of) document(s). Represent words by vectors in the d-dimensional space V_P, where the coordinate γ_i of word w is the frequency of co-occurrence with p_i. The projection onto the one-dimensional subspace of V_P generated by p_i is the vector $\gamma_i p_i$.

The scalar γ_i may be interpreted as the similarity of the word with p_i, but not as the probability that an arbitrary individual designated by w has

property p_i, because positive and negative occurrences like *some banks are safe*, *some banks are not safe* contribute both to γ_i. 'Reasoning by probability' based on frequency counts requires a distinction between positive and negative occurrences.

The opposite of a property implies with high probability the property. Intuitively, this explains why opposites are similar in frequency counts that do not distinguish between positive and negative occurrences.

6 Conclusion

New in this approach is that two separate notions of truth, one for concepts and one for sentences, are handled formally inside a single mathematical frame with a resulting equivalence of the two representations. The geometrical properties of quantum logic and the functional application of logic are preserved.

On a technical level, both the tensor product and syntactical analysis intervene when composing meanings.

Many interesting questions have not been addressed. For example, biproducts of concept spaces are necessary to handle predicates of an arbitrary arity simultaneously. Ambiguous words as well live in a biproduct of different concept spaces. Disambiguation by context uses the probability that the meaning factors through one branch rather than the other of the biproduct.

The most challenging questions belong to the probabilistic approach to natural language semantics and its relation to compositionality. How to distinguish between opposites? (The usual probabilistic approach confounds them.) How to capture the intuitive interaction of statistical learning of concepts and their logical use?

References

Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pages 415–425, 2004.

Jon Barwise and Robin Cooper. *Formal Semantics: the essential readings*, chapter Generalized Quantifiers and Natural Language, pages 76–125. Wiley-Blackwell, 2002.

Johan van Benthem and Kees Doets. *Handbook of Philosophical Logic*, chapter Higher-Order Logic, pages 275–329. Reidel Publishing Company, Dordrecht, 1983.

Stephen Clark and S. Pulman. Combining symbolic and distributional models of meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, 2007.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. A compositional distributional model of meaning. In W. Lawless P. Bruza and J. van Rijsbergen, editors, *Proceedings of Conference on Quantum Interactions*. University of Oxford, College Publications, 2008.

Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

Joachim Lambek. *Substructural Logics*, chapter From categorial grammar to bilinear logic, pages 207–237. Oxford University Press, 1993.

Joachim Lambek. Type grammar revisited. In Alain et al. Lecomte, editor, *Logical Aspects of Computational Linguistics*, volume 1582 of *LNAI*, pages 1–27, Heidelberg, 1999. Springer.

Anne Preller. Category theoretical semantics for pregroup grammars. In Philippe Blache and Edward Stabler, editors, *Logical Aspects of Computational Linguistics*, volume 3492 of *Lecture Notes in Artificial Intelligence*, pages 254–270, 2005.

Anne Preller. Toward discourse representation via pregroup grammars. *Journal of Logic, Language and Information*, 16:173–194, 2007. doi: http://dx.doi.org/10.1007/s10849-006-9033-y.

Anne Preller and Joachim Lambek. Free compact *2*-categories. *Mathematical Structures for Computer Sciences*, 17(1):1–32, 2007. doi: http://dx.doi.org/10.1017/S0960129506005901.

Anne Preller and Mehrnoosh Sadrzadeh. Semantic vector models and functional models for pregroup grammars. *Journal of Logic, Language and Information*, 20(4):419–423, 2011. doi: http://dx.doi.org/10.1007/s10849-011-9132-2.

C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004.

Peter Selinger. Dagger compact closed categories and completely positive maps (extended abstract). In *Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005*, pages 139–163, 2007. doi: http://dx.doi.org/10.1016/j.entcs.2006.12.018.

Paul Smolensky. Connectionism, constituency and language of thought. In Robert Cummins and Denise Dellarosa Cummins, editors, *Minds, Brains, and Computers*, pages 284–308, 1988.

Dominic Widdows. *Geometry and Meaning*. Number 172 in CSLI lecture notes. CSLI Publications, 2004.